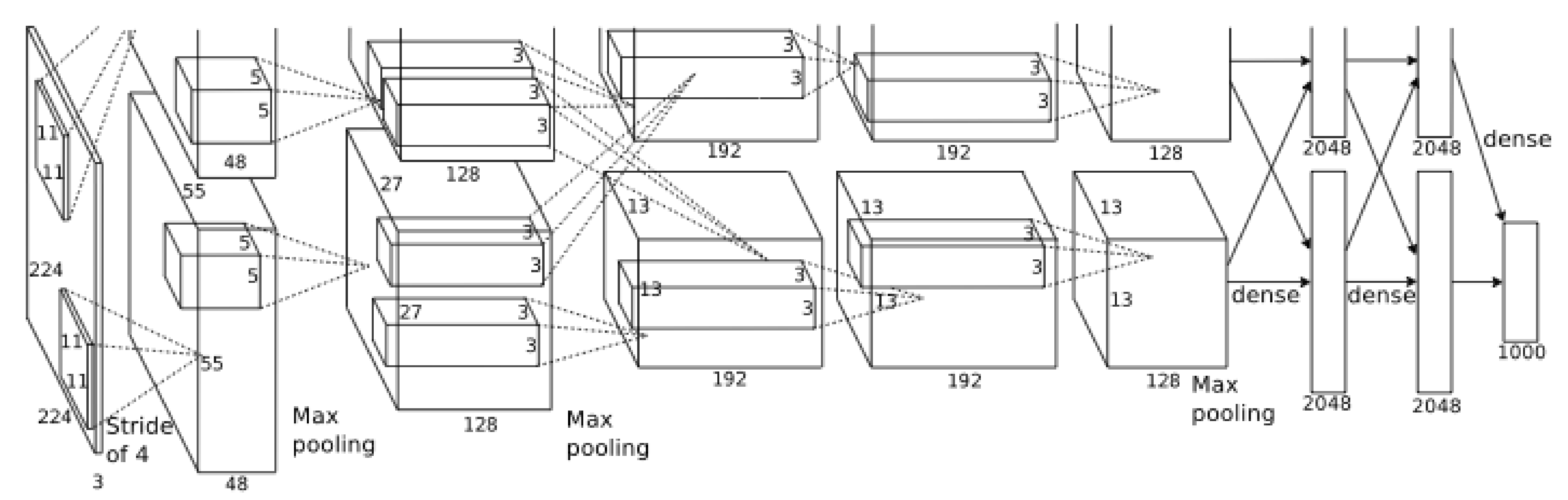
ENEE698A : Deep Learning Seminar

Spatial Pyramid Pooling in Deep Convolutional Networks for Visual recognition

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun

Raviteja Vemualapalli
November 13, 2014

Deep CNN

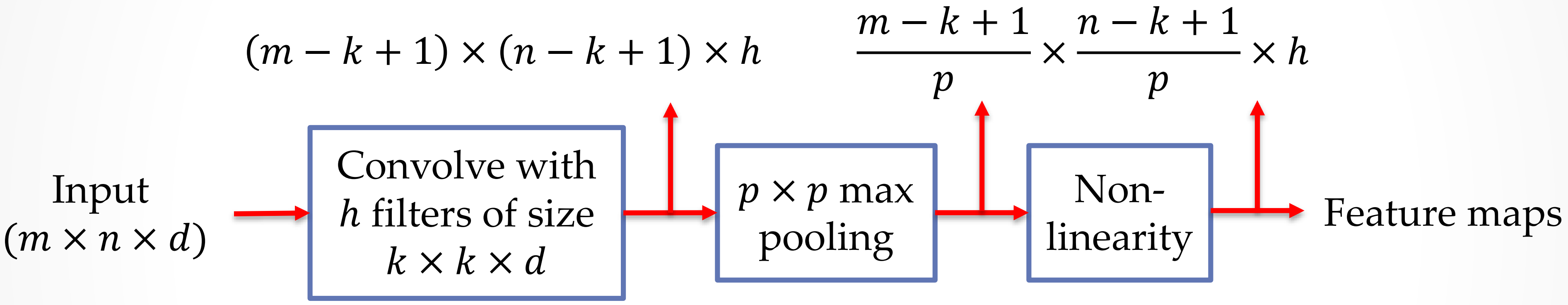


Input

Convolutional layers

Fully connected layers

Convolutional layer



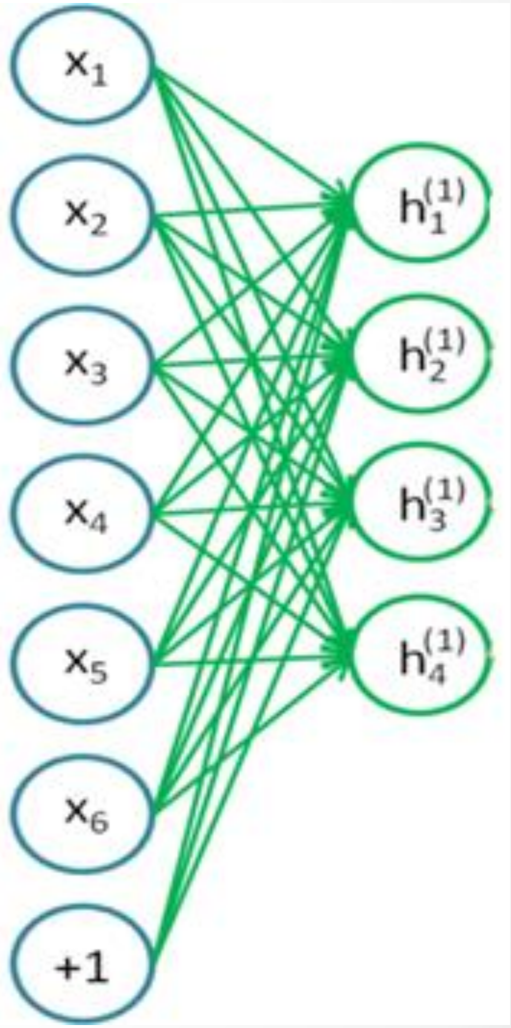
Size of the feature maps depends on the size of input for a given network.

Convolutional layers can be applied to input images of any size.

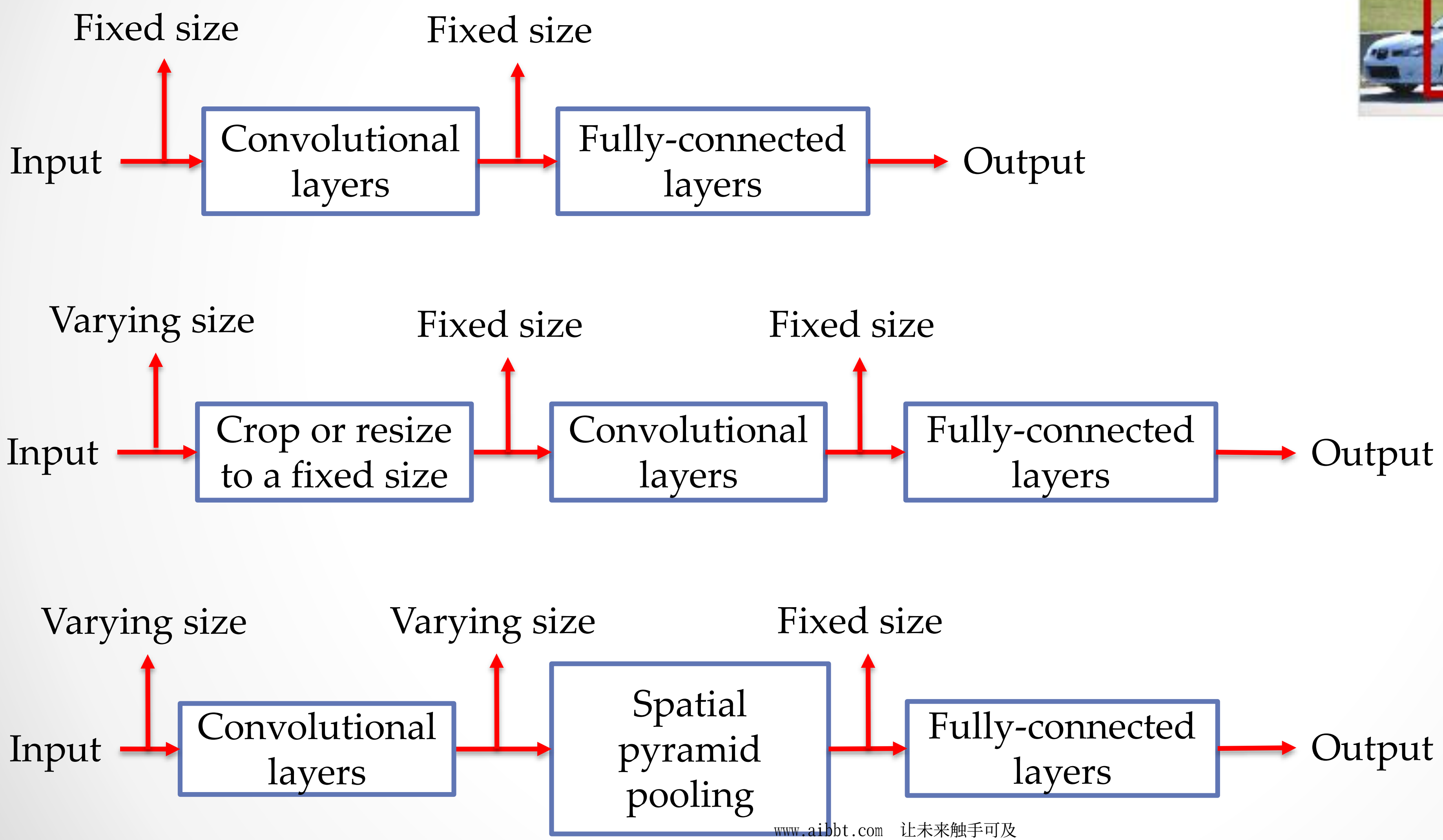
Fully connected layers

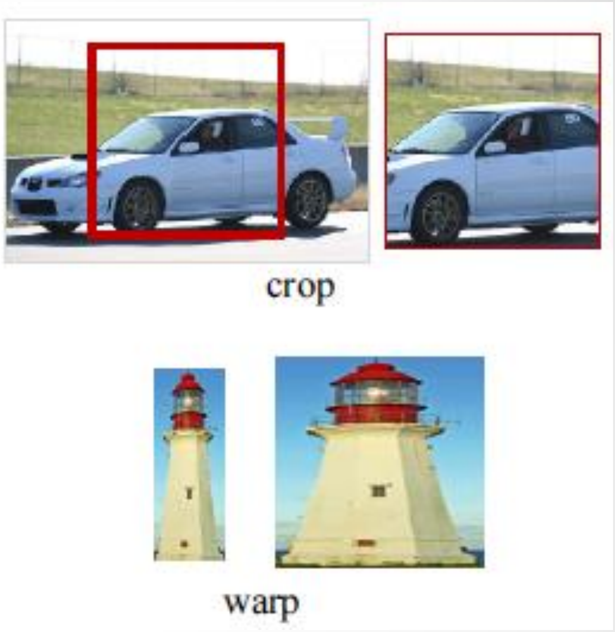
Fully connected layers require a fixed size input.

They cannot be applied to images of different sizes.



Size requirements





Pooling

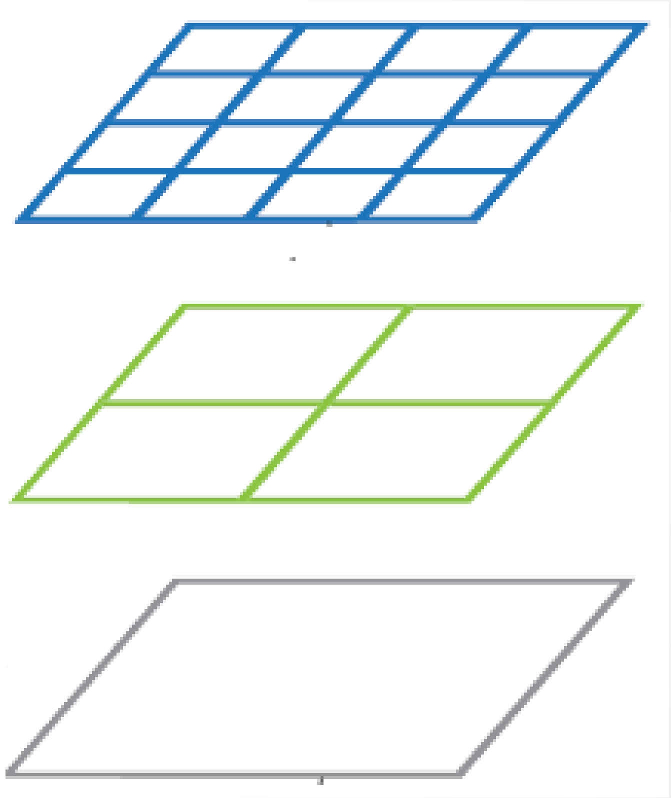
- Pooling function: Generates an aggregated representation for a set of features vectors $\left\{\overrightarrow{f_i}\right\}_{i=1}^{N}$.
  - Average pooling: $\frac{1}{N}\sum_{i=1}^{N}\overrightarrow{f_i}$.
  - Max pooling: Element-wise maximum
  - Second-order pooling: $\frac{1}{N}\sum_{i=1}^{N}\overrightarrow{f_i}\,\overrightarrow{f_i}^T$.

- The size of the pooling output does not depend on the number of features *N*.

Spatial pyramid pooling

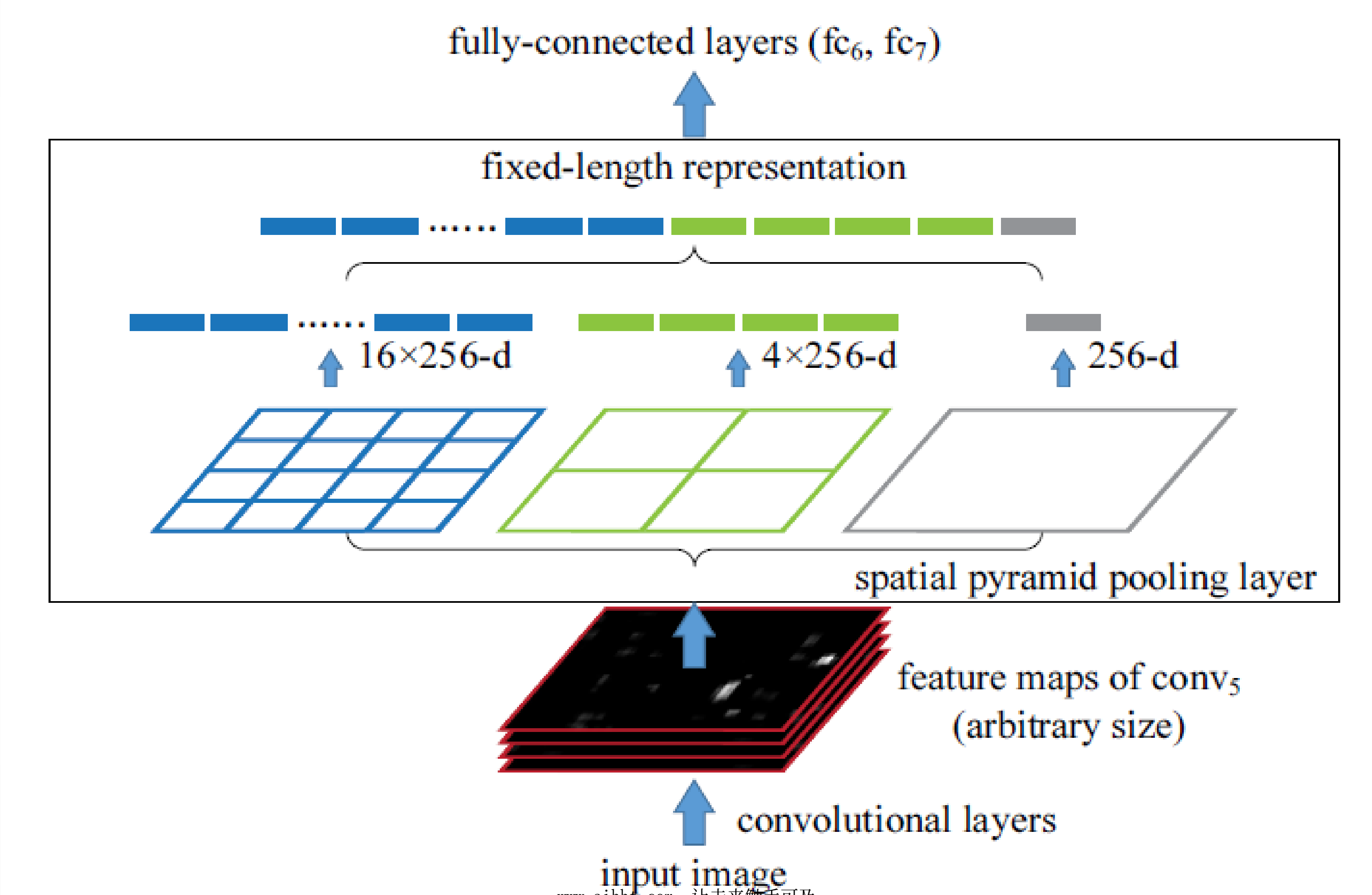
- Introduced in [Lazebnik 2006].

- Three steps:
  - Extract local feature descriptors at each pixel.
  - Divide the image into cells of different sizes.
  - Apply pooling function to each cell and concatenate all the pooling outputs.



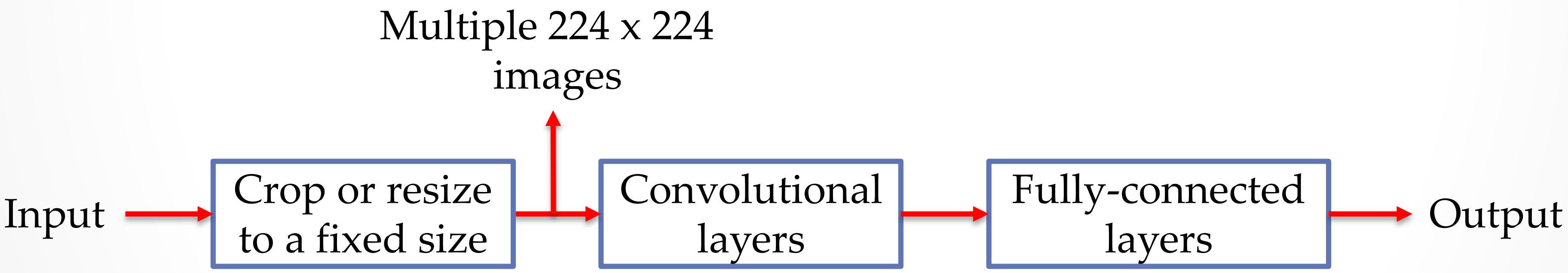
S. Lazebnik, C.Schmid, and J.Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories", CVPR 2006.

Spatial Pyramid Pooling in CNNs

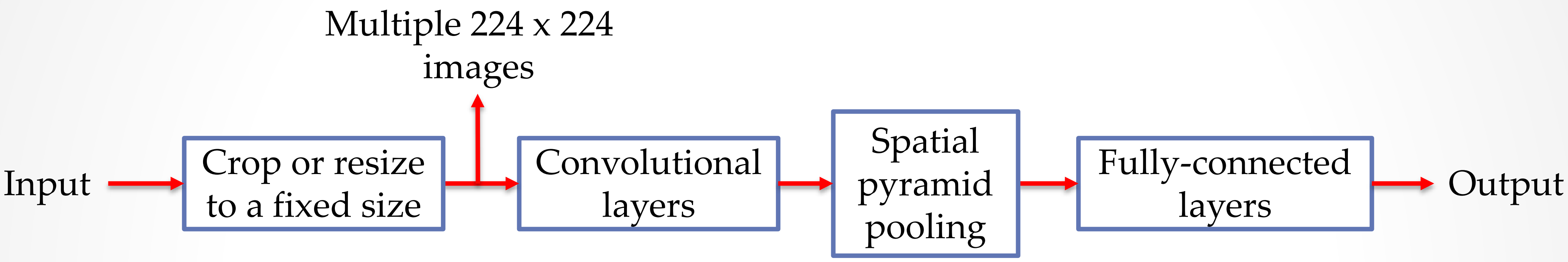


Experiments on ImageNet 2012

- Experimented with three different CNN architectures.
  - ZF-5 ([Zeiler 2013], 5 convolutional layers)
  - Convenet-5 ([Krizhevsky 2012], 5 convolutional layers)
  - Overfeat-5/7 ([Sermanet 2013], 5/7 convolutional layers)



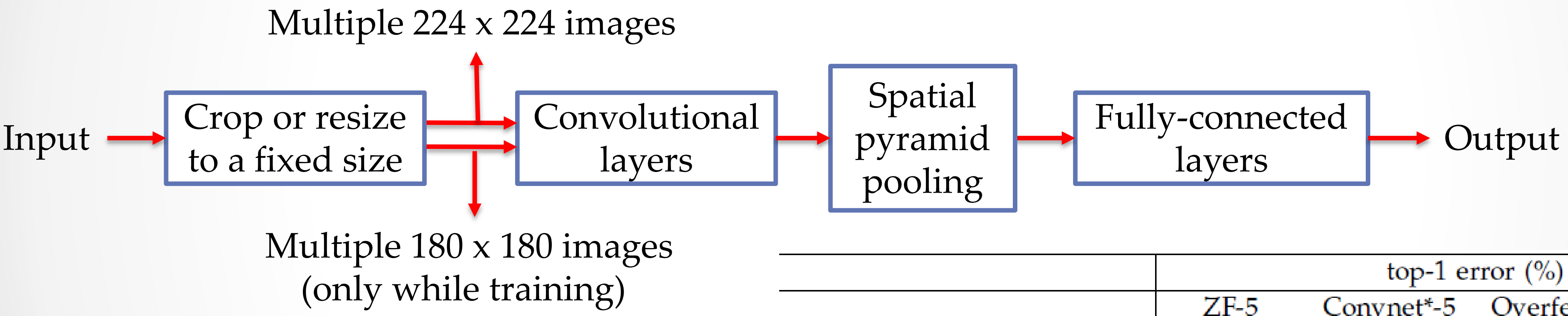
Spatial pyramid pooling improves accuracy



	top-1 error (%)			
	ZF-5	Convnet*-5	Overfeat-5	Overfeat-7
(a) no SPP	35.99	34.93	34.13	32.01
(b) SPP	34.98 (1.01)	34.38 (0.55)	32.87 (1.26)	30.36 (1.65)
	top-5 error (%)			
	ZF-5	Convnet*-5	Overfeat-5	Overfeat-7
(a) no SPP	14.76	13.92	13.52	11.97
(b) SPP	14.14 (0.62)	13.54 (0.38)	12.80 (0.72)	11.12 (0.85)

Multiscale training improves accuracy

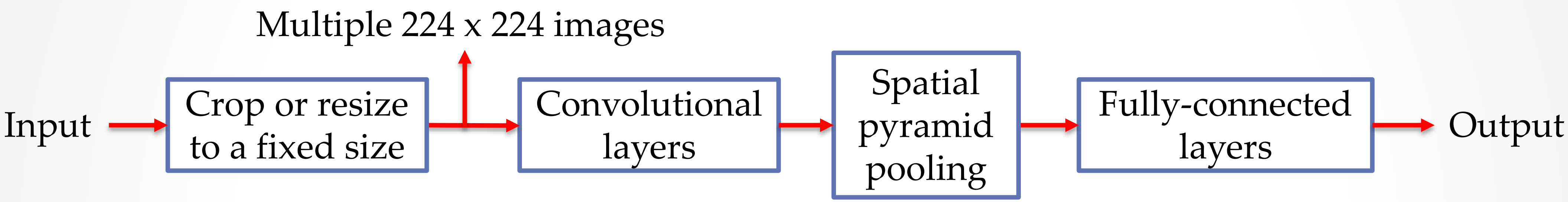
- Training with two sizes (224x224, 180x180), testing with 224x224 images.

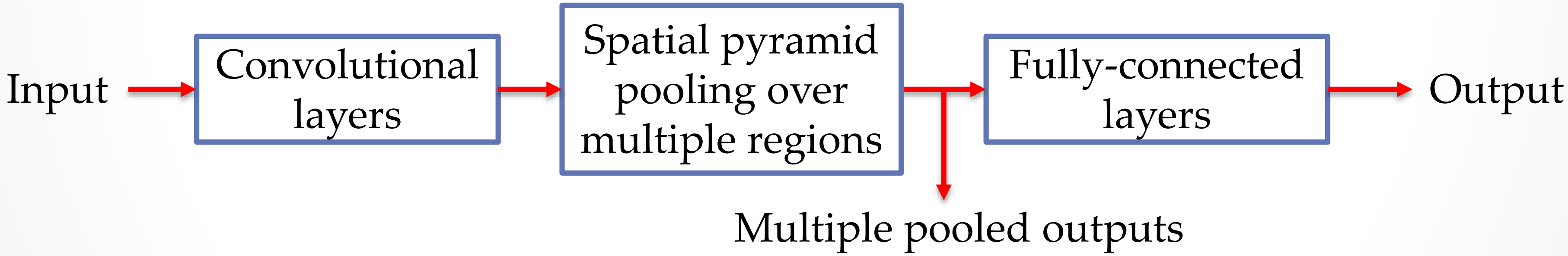


		top-1 error (%)			
		ZF-5	Convnet*-5	Overfeat-5	Overfeat-7
(a)	no SPP	35.99	34.93	34.13	32.01
(b)	SPP single-size trained	34.98 (1.01)	34.38 (0.55)	32.87 (1.26)	30.36 (1.65)
(c)	SPP multi-size trained	34.60 (1.39)	33.94 (0.99)	32.26 (1.87)	29.68 (2.33)

		top-5 error (%)			
		ZF-5	Convnet*-5	Overfeat-5	Overfeat-7
(a)	no SPP	14.76	13.92	13.52	11.97
(b)	SPP single-size trained	14.14 (0.62)	13.54 (0.38)	12.80 (0.72)	11.12 (0.85)
(c)	SPP multi-size trained	13.64 (1.12)	13.33 (0.59)	12.33 (1.19)	10.95 (1.02)

Reducing computation time





Much faster than applying convolutional layers to multiple images.

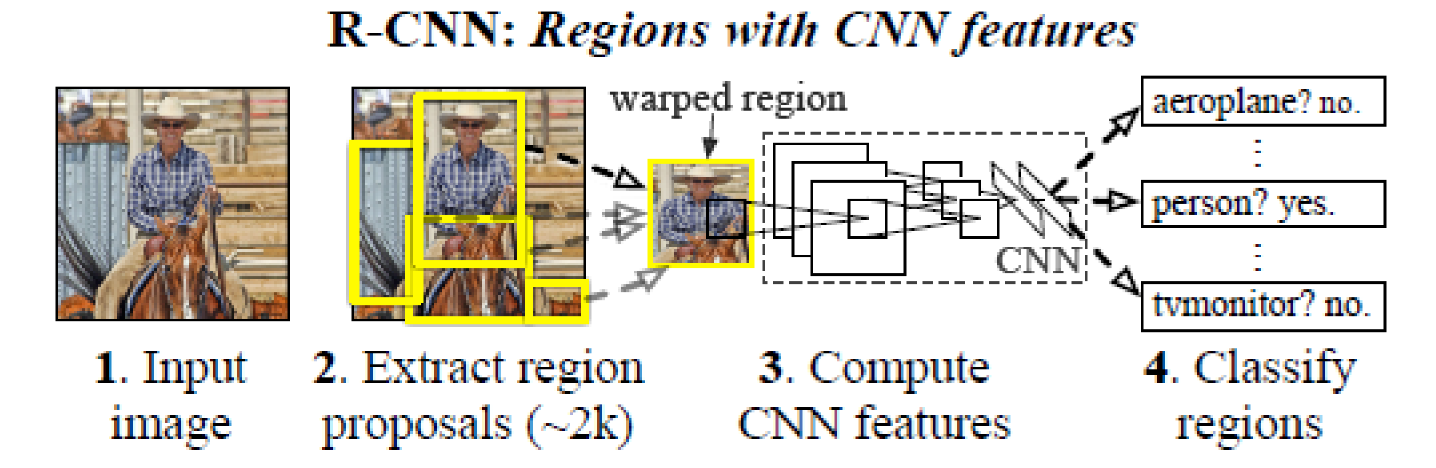
Multiscale network results

- Resized each image to six different scales.
- Applied CNN with SPP to six images.
- For each scale, SPP was applied to multiple regions in the final feature maps.
- A total of 98 different outputs were obtained from each image.
- Final result was based on the average of 98 outputs.

method	test scales	test views	top-1 val	top-5 val	top-5 test
Krizhevsky *et al.* [3]	1	10	40.7	18.2	
Overfeat (fast) [5]	1	-	39.01	16.97	
Overfeat (fast) [5]	6	-	38.12	16.27	
Overfeat (big) [5]	4	-	35.74	14.18	
Howard (base) [32]	3	162	37.0	15.8	
Howard (high-res) [32]	3	162	36.8	16.2	
Zeiler & Fergus (ZF) (fast) [4]	1	10	38.4	16.5	
Zeiler & Fergus (ZF) (big) [4]	1	10	37.5	16.0	
Chatfield *et al.* [6]	1	10	-	13.1	
ours	1	10	29.68	10.95	
ours	6	96+2full	**27.86**	**9.14**	**9.08**

Detection on PascalVOC 2007 using RCNN

- Generate 2000 object proposals using selective search.
- Resize each region into a pre-defined size (227x227).
- Extract features from each region using a deep CNN.
- Classify these features using SVM detectors.
- Runs CNN 2000 times.

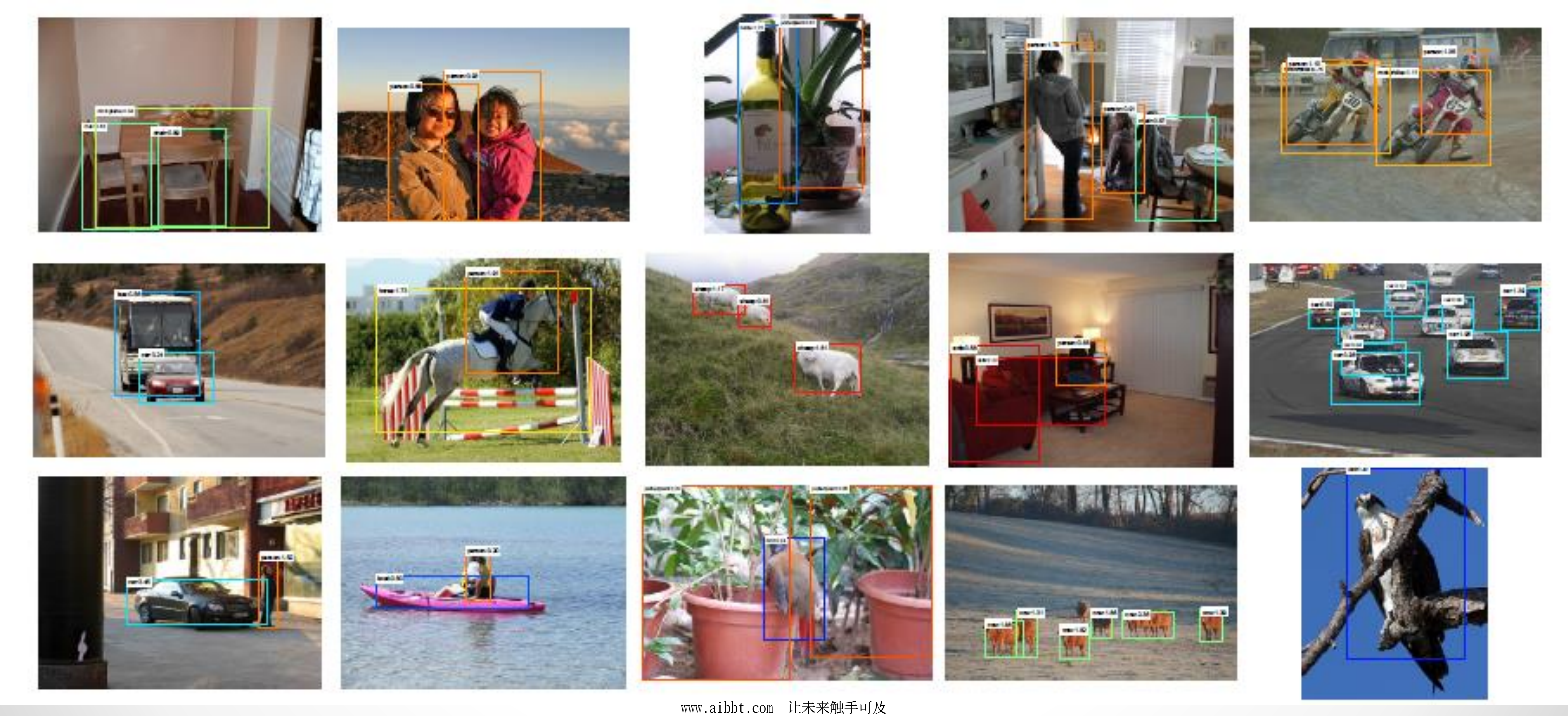


Detection using CNN+SPP

- Run convolutional layers on the entire image only once.
- Generate 2000 object proposals using selective search.
- Map each object proposal region in the input image to the corresponding region in the output of final convolutional layer.
- Use SPP to extract features from the final convolutional layer for each object proposal.
- Classify these features using SVM detectors.

mAP	58.0	58.5
conv time (GPU)	0.053s	8.96s
fc time (GPU)	0.089s	0.07s
total time (GPU)	0.142s	9.03s
speedup (*vs.* RCNN)	64×	-

Detection resutls



Thank You